

EVOLUCIÓN DE LA ASIGNATURA METODOLOGÍA Y TECNOLOGÍA DE LA PROGRAMACIÓN II

José Joaquín Aguilera¹, José Ramón Balsas¹, M^a José del Jesus¹

¹*Departamento de Informática. Escuela Politécnica Superior.
Universidad de Jaén. e-mail: {jjaguile, jrbalsas, mjjesus}@ujaen.es*

Resumen: La asignatura anual Metodología y Tecnología de la Programación de los estudios de I.T. en Informática de Gestión de la Universidad de Jaén ha evolucionado desde su implantación hasta convertirse en dos asignaturas cuatrimestrales. En este trabajo describimos una propuesta metodológica para la asignatura Metodología y Tecnología de la Programación II y presentamos las experiencias adquiridas en la docencia de esta materia.

1.- INTRODUCCIÓN

La asignatura Metodología y Tecnología de la Programación II (MTP II) pertenece al Plan de Estudios de 1997 de la Ingeniería Técnica en Informática de Gestión impartido en la Escuela Politécnica Superior de la Universidad de Jaén. Es una asignatura troncal del primer curso, con una carga docente de 3 créditos teóricos y 3 créditos prácticos, distribuidos en el segundo cuatrimestre. En el primer cuatrimestre del mismo curso se imparte la asignatura Metodología y Tecnología de la Programación I (MTP I) que establece conceptos previos y necesarios para la asignatura en la que se centra este trabajo.

En el Plan de Estudios de 1992 de la misma titulación, los contenidos de las asignaturas Metodología de la Programación I y II se incluían en una única asignatura troncal anual de 15 créditos (9 teóricos y 6 prácticos). El objetivo de esta asignatura era establecer las bases de los métodos y de las técnicas de la programación que permitiesen al alumno asimilar el resto de los conocimientos relativos a programación impartidos en asignaturas de cursos posteriores.

La experiencia obtenida con la docencia en la asignatura anual y en las nuevas cuatrimestrales nos ha permitido establecer la propuesta docente que presentamos a continuación.

2.- OBJETIVOS DE LA ASIGNATURA

Los objetivos de la asignatura MTP II son los siguientes:

- Profundizar en los métodos y técnicas de programación.
- Iniciar al alumno en el análisis de eficiencia de algoritmos y en el uso de métodos de prueba no formales.
- Comprender y aplicar algoritmos recursivos en la resolución de problemas.
- Conocer y analizar distintos algoritmos de ordenación y búsqueda y tener capacidad para adaptarlos eficientemente a situaciones concretas.
- Conocer y comprender los diferentes paradigmas de programación, para tener una visión más amplia de las distintas formas de implementar un algoritmo.

3.- EVOLUCIÓN DE LA ASIGNATURA

En el primer plan de estudios de Ingeniería Técnica en Informática de Gestión de la Universidad de Jaén (1992), se impartía en el primer curso la asignatura troncal de Metodología y Tecnología de la Programación (MTP), con carácter anual y una carga lectiva de 15 créditos (9 teóricos y 6 prácticos). En la reforma de dicho plan (1997) esta asignatura desapareció para adaptarse a la normativa del consejo de Universidades sobre el carácter cuatrimestral de las asignaturas.

Surgieron dos nuevas asignaturas: MTP I y MTP II. En un principio se pensó que cada una de ellas tuviese la mitad de la carga lectiva de MTP, 7.5 créditos. Finalmente se dejó la parte de introducción y fundamentos de programación (diseño de algoritmos y diseño de programas: descomposición modular y documentación) para MTP I con una carga lectiva de 9 créditos (6 teóricos y 3 prácticos) y el resto: conceptos avanzados de programación y técnicas complementarias de ayuda a la programación (análisis de algoritmos, lenguajes de programación, técnicas de verificación y prueba de programas) para MTP II con una carga lectiva de 6 créditos (3 teóricos y 3 prácticos).

El primer curso en el que se impartió esta nueva asignatura (1997/98), el temario teórico fue el siguiente:

1. Metodología de la Programación.
2. Análisis de Eficiencia de Algoritmos Iterativos.
3. Recursividad.
4. Algoritmos de Ordenación y Búsqueda.
5. Técnicas de Validación y Verificación de Programas.
6. Introducción a la Programación Orientada a Objetos.
7. Otras Técnicas de Programación.

El temario de las prácticas fue el siguiente:

1. Introducción a los Lenguajes de Programación Visual: Delphi.
2. Componentes y programación en Delphi.
3. Realización de un Proyecto de Gestión.

Se mantuvo la idea de la asignatura de MTP: los alumnos debían realizar un proyecto final de gestión en el que se contemplaran los objetivos de programación impartidos en la teoría. Se utilizó un lenguaje de Programación Visual, para así hacer más hincapié en el tema de “Otras Técnicas de Programación”. El lenguaje elegido fue Delphi, por ser Pascal el lenguaje empleado en la asignatura MTP I. Así los alumnos ya conocían gran parte de la sintaxis del lenguaje y sólo había que introducirles los nuevos conceptos de Programación Visual.

4.- ORGANIZACIÓN DE LA ASIGNATURA

Después de la experiencia obtenida impartiendo varios años la asignatura MTP y la experiencia y resultados obtenidos en el curso 97/98 en MTP II (gran número de alumnos que abandonaron la asignatura y alto índice de suspensos) se plantearon algunos cambios en la asignatura.

a) Teoría. Se introdujo el análisis y eficiencia de algoritmos recursivos, pues se vio que era necesario para poder comprender la eficiencia de los algoritmos recursivos de ordenación y búsqueda. Esta modificación hizo necesaria una reordenación temporal de los contenidos, anteponiendo el tema de Recursividad al de Análisis de Eficiencia.

Se eliminó el capítulo de Programación Orientada a Objetos por quedar incluidos sus contenidos en la asignatura Estructura de Datos y de la Información II del nuevo plan de estudios.

La tercera modificación fue adelantar el capítulo de Técnicas de Validación y Verificación de Programas para que, cuanto antes, los alumnos aplicasen los conocimientos adquiridos en él y pudiesen comprobar que sus programas son válidos y correctos.

El nuevo temario teórico, con su temporización, quedó como sigue:

1. Metodología de la Programación. (3 horas).
2. Técnicas de Validación y Verificación de Programas. (5 horas).
3. Recursividad. (3 horas).
4. Análisis de Eficiencia de Algoritmos. (5 horas).
5. Algoritmos de Ordenación y Búsqueda. (7 horas).
6. Otras Técnicas de Programación. (6 horas).

b) Prácticas. En el planteamiento inicial existía un cierto desequilibrio entre los contenidos teóricos y prácticos, debido a que la parte práctica se había convertido en una continuación de la asignatura MTP I.

En la nueva organización se ha tenido en cuenta el carácter práctico de la asignatura (3 créditos). El peso de las prácticas frente a la teoría lo hemos plasmado en la ponderación de la nota final de la asignatura, asignando a la nota de prácticas un 40% del total de la misma (4 puntos). Este porcentaje nos conduce a una planificación detenida de los contenidos para adecuarlos a los objetivos perseguidos en la teoría y para conseguir del alumno un alto grado de motivación, ya que de su correcta realización podrá obtener un elevado porcentaje de la nota final.

Las prácticas se realizan por parejas y están planificadas temporalmente para que todo el trabajo que deba realizar cada equipo pueda desarrollarse completamente en sesiones semanales de 2 horas asignadas a tal fin. No obstante el alumno tiene la posibilidad de preparar la práctica previamente a su realización utilizando el cuadernillo que se les entrega al principio del curso. En él se encuentra descrito el proceso a seguir para la realización de la práctica, así como los contenidos teóricos necesarios para su correcta realización y algunos ejemplos prácticos que les sirvan como referencia. Posteriormente aparecen descritos los problemas que se deben resolver y una lista de los contenidos que tendrán que entregar.

Para favorecer la labor continua del alumno en el desarrollo de las prácticas y dada la dificultad que supone para el profesor la corrección de las mismas por el elevado número de alumnos de la asignatura (317), se ha planificado un calendario de entrega. Las entregas se harán por bloques y cada uno estará formado por una o varias prácticas. La fecha de entrega no coincide con la realización de la última práctica del bloque, sino que se dan 2 semanas adicionales para que el alumno disponga de tiempo suficiente.

El temario de prácticas está formado por los siguientes puntos. Indicamos además el bloque, la temporización y las fechas de entrega de cada una de las prácticas, para el curso 98/99.

Bloque 1 (Entrega 23 Abril)

Práctica 1. Diseño de Unidades. Gestión de memoria dinámica. (3 sesiones)

Completar el código de una Unidad de listas doblemente enlazadas.

Práctica 2. Pruebas del Software (1 sesión)

Diseño de pruebas de caja blanca y caja negra para un conjunto de programas. Detección de errores y resolución.

Bloque 2 (Entrega 28 Mayo)

Práctica 3. Recursividad (2 sesiones)

Funciones *par e impar* mutuamente recursivas. Las torres de Hanoi
El problema de las 8 reinas.

Práctica 4. Algoritmos de Búsqueda (2 sesiones)

Búsqueda de patrones en una cadenas: algoritmo de Boyer-Moore

Bloque 3 (Entrega 16 Junio)

Práctica 5. Algoritmos de Ordenación (3 sesiones)

Implementación del algoritmo QuickSort. Cálculo del umbral. Elección del algoritmo base. Función de tiempos. Análisis de eficiencia.

Algunas prácticas disponen de apartados opcionales que permiten a los alumnos que lo deseen aumentar la calificación final obtenida.

La experiencia nos había demostrado que el alumno tiende a abandonar la asignatura en los primeros meses ante el esfuerzo que supone tener que entregar unas prácticas periódicamente, unido a la presión que ejerce la carga del resto de asignaturas. Para aumentar la motivación y evitar el abandono de la asignatura, la entrega de cada uno de los bloques de prácticas no es rígida y si se realiza en la fecha asignada supone una puntuación de las mismas sobre 12 puntos en lugar de sobre 10. Si el bloque no se entrega en su plazo se podrá entregar el día del examen, pero sin bonificación.

Periódicamente se colocan en el tablón de anuncios las notas, lo que permite a los alumnos realizar un seguimiento de sus resultados y conocer los fallos cometidos (a través de las tutorías) con el fin de no volver a repetirlos en las siguientes entregas.

5.- CONCLUSIONES

La experiencia obtenida durante el curso 98/99 nos ha permitido extraer las siguientes conclusiones:

- La nueva temporización de los contenidos teóricos ha hecho que el alumno tenga una visión más compacta de la asignatura y ha ayudado a un mejor seguimiento de las prácticas.

- La inclusión del análisis de eficiencia de algoritmos recursivos ha permitido al alumno una mayor comprensión de los algoritmos de ordenación y búsqueda, despertando un mayor interés por este tema.
- Al continuar con el lenguaje de programación visto durante el primer cuatrimestre, conseguimos que afiancen sus conocimientos de programación, a la vez que nos permite centrarnos más detalladamente en los contenidos vistos en teoría.
- La flexibilidad conseguida con las prácticas ha permitido al alumno adaptarse gradualmente a su desarrollo con lo que se ha despertado el interés general y hemos conseguido un incremento importante de participación y resultados frente a cursos anteriores.
- La sincronización de los temarios de prácticas con los de teoría nos ha permitido potenciar el grado de asimilación de los conceptos aprendidos. Este hecho supone para el alumno una reducción de la complejidad de la asignatura evitando con ello la tentación de abandonar la asignatura.
- Se ha notado un aumento notable de la asistencia a tutorías motivada por la evaluación continua de las prácticas.

Entre nuestros objetivos futuros están la obtención de enunciados de problemas más atractivos capaces de atraer más la atención del alumno y la posibilidad de aumentar el porcentaje de nota asignada a la parte práctica si conseguimos mantener el interés demostrado durante este año por los alumnos en el desarrollo de las prácticas.

6.- BIBLIOGRAFÍA

La bibliografía principal propuesta para la materia a impartir es la siguiente:

- [1] Cortijo, F.J.; Cubero, J.C.; Pons,O. (1993). Metodología de la Programación. Programación y Estructuras de Datos en Pascal. Proyecto Sur
- [2] Joyanes Aguilar, L. (1996). Fundamentos de Programación: Algoritmos y Estructuras de Datos. (2ª Edición). McGraw-Hill.
- [3] Balcázar, J.L (1993). Programación Metódica. McGraw-Hill.
- [4] Cueva, J.M.; García, Mª P.A. y otros (1994). Introducción a la Programación Estructurada y orientada a objetos en Pascal. Impreso en Gráficas Oviedo, S.A.
- [5] Dale, N.; Lilly,S.C. (1986).Pascal y Estructuras de Datos. McGraw-Hill.
- [6] Pressman, R.S.; (1997). Ingeniería del Software. Un Enfoque Práctico. 4ª Edición. McGraw-Hill.
- [7] Brassard,G.;Bradley,P.(1997).Fundamentos de Algoritmia. Prentice-Hall

- [8] Knuth, D.E. (1995). El arte de programar ordenadores. Vol.1: Algoritmos fundamentales y Vol. 3: Clasificación y Búsqueda. Reverté.
- [9] Bal, H.E.; Grune, D. (1994). Programming Language Essentials. Addison-Wesley.